Lab 1

Problem 1. Which of the following functions are increasing? eventually nondecreasing? If you remember techniques from calculus, you can make use of those.

a. $f(x) = -x^2$
b. $f(x) = x^2 + 2x + 1$
c. $f(x) = x^3 + x$

Problem 2. Use the limit definitions of complexity classes given in class to decide whether each of the following is true or false, and in each case, prove your answer.

a. $4n^3 + n$ is $\Theta(n^3)$.
b. $\log n$ is $o(n)$.
c. 2^n is $\omega(n^2)$.
d. 2^n is $o(3^n)$.

Problem 3. Show that for all $n > 4$, $2^n < n!$. Hint: Use induction.

For problems below, write Java programs that solve these problems as efficiently as possible.

Problem 4. GCD Problem: Given two positive integers m, n, is there a positive integer d that is a factor of both m and n and that is bigger than or equal to every integer d' that is also a factor of m and n?

Write a Java method int gcd(int m, int n) which accepts positive integer inputs m; n and outputs the greatest common divisor of m and n.

Examples

- If m = 12 and n = 42, return 6
- If m = 7 and n = 9, return 1

Problem 5. Implement the following Java method.

```
public static int secondSmallest(int[] arr) {
        if(arr==null || arr.length < 2) {
                throw new IllegalArgumentException("Input array too small");
        }
        //implement
}
```

This method returns the second smallest element of the input array.

Examples

- If input is [1, 4, 2, 3], return 2.
- If input is [3, 3, 4, 7], return 3. (Smallest is 3, and second smallest is 3.)
- If input is [9], your program will throw an exception.

Problem 6. SubsetSum Problem: given a set S = {s_0, s_1,s_2, …, s_{n-1}}of positive integers and a non-negative integer k, is there a subset T of S so that the sum of the integers in T equals k?

Formulate your own procedure for solving the SubsetSum. Think of it as a Java method subsetsum that accepts input S and k, and outputs a subset T of S with the property that the sum of the elements in T is k if such a T exists, or null if no such T can be found.

Examples

- If S is [1, 3, 9, 4, 8, 5] and k = 21, return [9,4,8] (since 9 + 4 + 8 = 21)
- If S is [1, 3, 9] and k = 5, return null (since no such subset can be found)
- If S is [1, 3, 9, 4, 8, 5] and k = 0, return [] (since the sum of the empty set is 0)